第4章 フル・トレース，外部バス・トレース，オンチップ・トレースの違い

ICEによるトレース機能を使ったデバッグ手法

大垣 憲和

ICEを使ったデバッグ機能の一つとしてトレースがある．トレース機能は，ブレーク機能を使ったデバッグと比較して，マイコンを止めずに動いた状態でデバッグ・ログを採取できる．
ここでは，トレース機能を実現するためのハードウェアとその手法について解説する．（編集部）

1．トレース機能とは

● トレースとブレークの違い

トレースは，ICE（In-Circuit Emulator）のみが実現できる重要な機能です．これに対してICEにはブレーク機能もあります．ブレーク機能はマイコンを一時停止させ，その時点でのレジスタやメモリの値を参照するものです．しかし，各種センサやモータ制御のように，マイコンが周辺機器をリアルタイム制御するようなアプリケーションではマイコンを停止させることはできません．このようなタイミングの厳しいプログラムのデバッグには，マイコンを動作させた状態で実行履歴を記録・分析できる，トレース機能が必要になります．

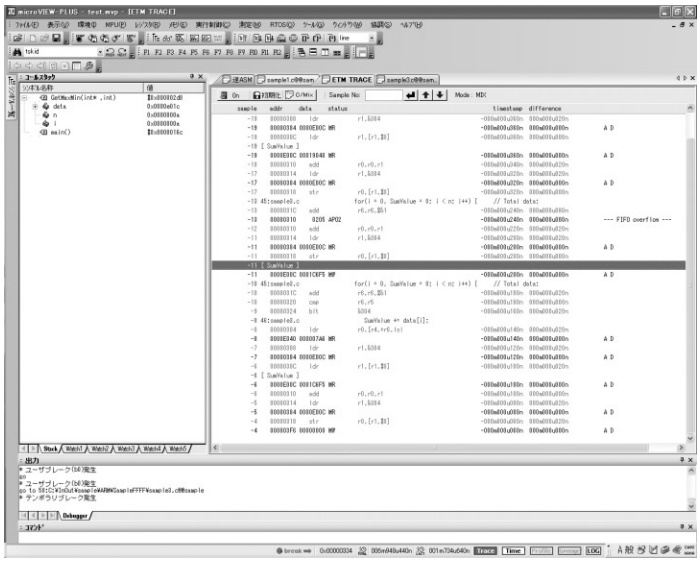

図1　トレース解析画面例

● トレースは実行タイミングに影響を与えない

トレースは，マイコンが実行するすべての命令実行とデータ・アクセスを記録できます．そのため，マイコンの実行タイミングに影響を与えない唯一のリアルタイム・デバッグ手法といえます．

リアルタイム制御以外のアプリケーションでも，「突然プログラムが想定外の動きをしてしまう」ような複雑な事象のデバッグには，トレースを使用してマイコンの実行履歴をさかのぼって原因解析を行います．再現性の乏しい不具合のデバッグにはトレース機能が不可欠です．

● printfもデバッグ手法の一つ

トレース以外の動的解析手法として，プログラム中にデバッグ・コンソールへのメッセージ出力を組み込むものがあります．この方法はC言語ソース・コード中に`printf`を組み込むことから「printfデバッグ」と呼ばれます．

この手法はマイコンの占有時間が長く，メッセージを挿入した場合としない場合で周辺機器の制御タイミングが大きく変化してしまうため，リアルタイム制御プログラムのデバッグには使用できません．また，printfデバッグは挿入した部分だけのメッセージが記録されるので，不具合原因の特定ができないケースでは，`printf`文を挿入し再コンパイル，再テスト，再コンパイル…と，何度もこの作業を繰り返すことになります．トレースを使用し実行履歴の解析を行えば，このような無駄なデバッグ・サイクルを削減でき，デバッグ時間の大幅な効率化が実現できます．

KEYWORD ── トレース機能，ICE，フル・トレース，外部バス・トレース，オンチップ・トレース